# *Networked Avionics Part Task Trainer*

**NAVY SBIR 99-185, Phase I**
**N68335-00-C-0143**

**Item: 0001AF**

***Final Report***

**12/12/99 – 6/9/00**

***AAC, Inc.***
**12140 Double Tree Lane**
**Lusby, MD 20657**

**410-326-5061**

**Prepared By: Robert B. Andrews**

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From – To)* |
|---|---|---|
| 09-06-2000 | SBIR PHASE I - FINAL REPORT | Dec 1999 – Jun 2000 |

**4. TITLE AND SUBTITLE**

US NAVY SBIR N99-185
Networked Avionics Part Task Trainer

**5a. CONTRACT NUMBER**
N68335-00-C-0143

**5b. GRANT NUMBER**
N/A

**5c. PROGRAM ELEMENT NUMBER**
0001-AF

**6. AUTHOR(S)**

Andrews, Robert B.

**5d. PROJECT NUMBER**
N/A

**5e. TASK NUMBER**
N/A

**5f. WORK UNIT NUMBER**
N/A

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

AAC, Inc.
12140 Double Tree Lane
Lusby, MD 20657

**8. PERFORMING ORGANIZATION REPORT NUMBER**
N/A

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Commanding Officer, Indian Head Division
Attn: T. Margeson Code 8210C
101 Strauss Ave
Naval Surface Warfare Center
Indian Head, MD 20640-5035

**10. SPONSOR/MONITOR'S ACRONYM(S)**
NAVAIRSYSCOM, PMA205

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER**
0001-AF

**12. DISTRIBUTION AVAILABILITY STATEMENT**

Approved for public release; SBIR report, distribution unlimited

**13. SUPPLEMENTARY NOTES**

None

**14. ABSTRACT**

The purpose of this work was to develop a conceptual design for enabling networking and collaborative training using Avionics Part Task Trainers (APTTs). R&D efforts resulted in a conceptual design comprising an innovative adaptation and utilization of the DoD High Level Architecture (HLA). AAC defined design requirements for a Software Development Kit (SDK) that provides APTT application developers an "export C" Application Programming Interface (API) enabling functional HLA integration for applications developed in Visual Basic, Authorware, and Rapid CBT. Results of this effort include a high level design capable of supporting prototype development of a distributed set of APTT applications executing in an HLA federation. Potential application of the SDK product applies to any PC based application that can benefit from, or has growth plans to, provide distributed data sharing over the Internet. This could include multiple student PC workstations, Instructor PC workstations, and monitoring server applications.

**15. SUBJECT TERMS**

CDNU, Emulation, Trainer, Avionics

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Robert B. Andrews |
| U | U | U | UU | 23 | 19b. TELEPHONE NUMBER *(Include area code)* 401.326.5061 |

STANDARD FORM 298 (Rev 8-98)
Prescribed by ANSI Std Z39-18

# Table of Contents

# 1 Introduction

Under the Navy SBIR 99-185 Phase I program, AAC conducted a comprehensive study to develop requirements and an innovative preliminary design to provide networking functionality to existing and future Avionics Part Task Trainers (APTTs). The proposed product is a unique Software Development Kit (SDK) for use with commonly used Integrated Development Environments (IDE's) like Visual Basic, C++, Rapid CBT, and Authorware. The SDK enables APTT applicaiton developers to modify or create new APTT applications that have High Level Architecture (HLA) compliant networking functionality.

Most APTTs are stand-alone (PC) computer systems hosting a variety of applications focusing on aircraft avionics training. A given APTT can host Simulation or Emulation and/or Computer Based Training (CBT) type applications developed by multiple vendors using a vareity of IDEs. APTT applications include active graphical emulations of avionics control panels, and user interactive drill response CBT programs.

AAC's proposed SDK product presents to the APTT application developer a high level Application Programming Interface (API), enabling efficient code development to support networked connectivity to distributed APTT applications having the same capability. Moreover, the SDK enables the developer to produce the needed code with a variety of Integrated Development Environments (IDEs). For example, a C++ IDE would utilize the SDK Class based DLLs, a Rapid CBT or Authorware IDE would use the "export C" based DLLs, while a Visual Basic IDE would use the ActiveX objects.

Concurrently with SDK design development, AAC performed market analysis of exiting technologies to substantiate currently perceived requirements. Initially, no SDKs could be found that met the requirements networked connectivity of APTT type applications, so AAC pursued SDK development building on the existing Microsoft system level Winsock DLL. A three Build approach defined the process plan to achieve a prototype design for this SDK.

Market analysis uncovered a system called OpenSkies developed by Cybernet Corporation during month three of Phase I performance. OpenSkies is a flight simulation product that supports networking between distributed users. The OpenSkies software product is High Level Architecture (HLA) compliant, which is mandated by DoD for simulation development. HLA provides a networking infrastructure to support TCP/IP distributed colaboration between PC hosted simulations, and is believed configurable to support networking between APTT type applications.

Since the APTT hosts subsystem simulations, emulations and CBT type applications, AAC began applying the HLA Federate Development and Execution Process (FEDEP) Model to APTT type applications. Should this prove successful, installing an HLA Run Time Infrastructure (RTI) on PC based APTTs, and imposing APTT application

requirements to function as HLA Federates, the topic's networking objectives will be met.

With the discovered HLA DoD mandate for simulations, AAC's topic solution shifted from the proposed plan to design from scratch (in Builds) a networking SDK based on the Winsock DLL, to utilizing existing PC hosted HLA compliant software provided by DMSO and key OpenSkies components to meet the requirements for a Networked Avionics Part Task Trainer.

Specific areas investigated include:

1. Hosting an HLA RTI on APTT platforms (PCs).

2. Upgrading exiting (and future) APTT applications to HLA federates.

3. Integrating exiting and future APTT applications to the HLA RTI.

AAC has performed marketing activities during Phase I to present this planned capability to both commercial and government entities. There has been considerable interest in this product as it meets current needs of both market areas. However, R&D financial backing was not realized.

US Airways demonstrated interest as their Rapid CBT developed simulations are stand-alone applications resident on multiple PC's. Each stand-alone system requires a full duplicate database from which the simulation interface. This SDK will provide the means to retain one database where by all joined applications (users) can access. Software created from this product SDK will provide, through the use of an instructor application, a means for single point control of all "joined" simulations. Instructors will be able to insert system faults, and provide functional demonstrations that impact all "joined" applications (users) concurrently.

The government has demonstrated an interest in this product to support training of current communication related avionics capability. Specifically the ARC-210 Electronic Remote Fill, and COMSEC associated processes required by multiple aircraft radios.

## 2 Technical Objective

The foreseen cost and training associated with Avionics Part Task Trainers (APTTs) can be significantly improved by incorporating Internet based technologies. The resulting network functionality will enable collaborative aircrew-to-aircrew and aircrew-to-instructor training using currently stand-alone APTT systems.

The specific technical objective is to develop a network interface for APTT applications to perform embedded (background) data messaging between distributed APTT applications over the Internet.

There is a pronounced need to expand current APTT functionality for advanced communications and navigation training. Implementing functionality associated with this objective will reduce training costs related aircraft flight hours currently needed to achieve required multi aircrew readiness levels. Using the Internet to initiate avionics faults by way of a remote instructor application adds beneficial training capability to the APTT training mission.

The goals of the project are to determine the feasibility of a preliminary design to provide an Internet based software infrastructure to support collaborative training relevant to the APTT mission. This network interface is planned to be a marketable Software Development Kit (SDK) containing development tools like Dynamic Linked Libraries (DLLs) and ActiveX controls that provide a high level Application Programming Interface (API) that is High Level Architecture (HLA) compliant for the APTT application developer.

The overlying Phase I objective included in-depth analysis and proof of concept determination that an APTT application can be cost effectively implemented to perform background TCP/IP Internet data transfers transparent to the user. The aircrew interface must remain aircraft avionics representative during all modes of training operations.

Questions to be addressed include:

1. Is the application processing associated with background Internet message transfers to a server user detectable, and does this added processing result in unacceptable user interface degradation as compared to existing stand-alone operation?

2. Does Commercial-Off-The-Shelf (COTS) software exist that can provide the desired functionality and interfaces for the proposed design configuration?

3. Is there sufficient interest to pursue marketing Networked APTTs in the commercial market?

## 3 Approach Taken

AAC's approach to the technical objectives involved concurrent work in the following areas:

1) Requirements analysis.
2) Commercial-Off-The-Shelf (COTS) application research.
3) Design Development.
4) Marketing/Commercialization Approach

### *1.1 Requirements Analysis*

The requirements definition approach was not to freeze perceived requirements at initiation of (or early in) Phase I execution, but to pursue requirements definition through out Phase I concurrently with conceptual design development based on an identified set of requirements at the time. This injects the potential re-design of the conceptual design at any point in time, as requirements remain fluid. New, changing, or deleted requirements impact the current state of the conceptual design, which make the design a moving target, often requiring a modified or new conceptual design approach.

Midway through Phase I execution, requirements analysis, fed by concurrent COTS research, led to a major redesign of the Networked APTT resulting in the abandonment of the initial conceptual design approach. With the discovery of the HLA compliant OpenSkies application, and subsequent establishment of non-disclosure agreements with Cybernet (developers of OpenSkies), AAC realigned design requirements to include derived requirements associated consistent with the integration of OpenSkies components (Networking and Recording DLLs) into the product design.

Defining the process of modifying an APTT application to function like an HLA Simulation or Federate provided the mechanism to identify system requirements. Many design requirements, and following design decisions, are based on derived requirements from the OpenSkies components. AAC has not received interface details on these OpenSkies components, which has limited AAC's ability to independently pursue a prototype design.

Researching DMSO provided HLA documentation and independently studying Cybernet's OpenSkies Networking and Recording DLLs (using Microsoft's Dependency Walker application), believed valid assumptions pursuant to the OpenSkies DLL interface requirements were derived.

Identification and partitioning of functional requirements to HLA objects is essential prior to initiating design work. Adaptation of the HLA SOM/FOM development process correlates to the development process of networking APTT applications, and has been the model providing the basis for this effort.

### *1.2 COTS Application Research*

To substantiate an innovative conceptual design, an investigative search for existing software applications relevant to the technical objectives was initiated upon contract award. The potential of discovering existing COTS software that meets the topic objectives was (and is) seen as value added to both the government and to AAC.

Performing COTS research ensures that AAC does not (at undue cost to the government) "re-invent the wheel".

Early in the 3rd month of Phase I performance, the OpenSkies PC based Virtual Environment Training System developed by Cybernet Systems Corporation was discovered at a NAVAIRSYSCOM SBIR exposition. The OpenSkies COTS product provides the functional capability needed to meet the networking objectives for this topic.

Resulting from the OpenSkies discovery, research shifted from the current design of a build from scratch Winsock API implementation to a design based on the OpenSkies system. A Non Disclosure Agreement was established between AAC and Cybernet to facilitate data exchanges to confirm or deny the applicability of OpenSkies to a Networked APTT system. Research confirmed that OpenSkies is more than a feasible solution to addressing the technical objectives of this SBIR topic.

The OpenSkies system implements a lightweight HLA RTI that minimizes data throughput while maximizing user application performance. The OpenSkies RTI is very similar to MAK Technologies PC based distributed tank simulation in that it runs "real time" not using the HLA Routing Space functionality, which filters and prioritizes federate interactions.

Detailed research of the OpenSkies system is needed to integrate its Networking and Recording DLLs into a APTT_HLA DLL. AAC collaboration with Cybernet to complete this analysis is expected when contracting funding supports their active participation.

## *3.1 Design Development*

### 3.1.1 Initial Design Approach

The initial Phase I conceptual design approach defined a Networked APTT interface built from scratch upon Microsoft's Winsock API. The conceived product comprised a mixture of Commercial Off-The-Shelf (COTS) C/C++ software, and innovative custom software to create a system of networked APTT client applications, an instructor application, and a server application over the Internet. The applications utilize product Dynamic Linked Libraries (DLLs) insuring flexibility, adaptability, and product reuse with multiple avionics emulations (e.g., Multi-Function Displays, Heads Up Displays, and Control Panels) resident in multiple aircraft platforms (e.g., F-16, F-18, V-22, ATF, and Commercial Aircraft). The instructor application supports controlled asynchronous fault insertion into targeted APTT user systems for degraded avionics training. A full analysis, design, proof of concept demonstrator, and final report will be provided under Phase I. A validation demonstrator using the AH-1W Control Display Navigation Unit Emulation (CDNUE) will be provided as a Phase I option.

The initial conceptual design evolves into an SDK enabling modifications to an existing APTT User Application and designing four new applications in a spiral build software development process comprising three (3) builds. At the start of each new build, functionality for the new build is defined based on the perceived requirements at the time, which comprise the new build's functional requirements. The next phase of each build is to define the system performance requirements that will support the functional requirements. This is followed by the development of a software design that meets the defined performance requirements. Coding a representative prototype in software then validates the design. As each build progresses, an executable Networked APTT prototype emerges. Each proposed build addresses increasing functionality from the most basic client server connectivity to a final design that feasibly supports Phase II implementation.

Functionality identified in this initial design is achievable through innovative adaptation of the HLA compliant COTS application OpenSkies. Hence building a non-HLA compliant architecture from scratch is not a feasible or advantageous approach. The initial design development effort is no long required in that the desired networking functionality has been implemented in the OpenSkies software. Work associated with this initial design has been helpful in disclosing the requirements for a Networked Avionics Part Task Trainer. These requirements were recognized as being implemented in OpenSkies. This lead to the solution migration to OpenSkies versus creation of new software from scratch to meet the same objective.

### 3.1.2 Current Design Approach

HLA is mandated by DoD for Simulation development to enable code reuse and distributed networked interplay. HLA is not mandated for APTT type applications. However, remotely interfacing APTT applications are not that different from remotely interfacing simulations – they both require the same services. Moreover, APTT applications can make use of the HLA infrastructure to provide data sharing and sourcing between remote applications. HLA is a freebee that cannot be ignored - its directly applicable to the topic objectives. Hence, the design approach will incorporate HLA as implemented in the OpenSkies application.
The OpenSkies product encapsulates the HLA compliant "plumbing" for distributed users of the OpenSkies system. Incorporation of "already developed" OpenSkies network related components into the Networked APTT SDK, ensures that vendors of APTT applications developed HLA capable products. Moreover, the use of OpenSkies greatly reduces the development time of getting this SDK and Networked APTTs into the market place.

The OpenSkies software system meets and exceeds functionality previously planned with a scratch built SDK based on the initially proposed design concept. Instead of integrating APTT applications to a network API developed from scratch, an API based on OpenSkies components is the chosen current design approach. The OpenSkies Network and Recording DLLs are two key components that are needed to provide the

API to HLA services. AAC will integrate these two DLLs into one DLL adding needed functionality specific to APTT distributed processing. This new DLL has been named the APTT_HLA DLL.

Because APTT applications are not all developed in C++, many APTT applications cannot support DLL call back processing. Moreover, most APTT applications are limited to DLL "export C" function calls. A helper file, configured for a specific APTT application by the application developer, will provide data buffering and "call back" processing required to interface with the current OpenSkies Network and Recording DLLs. Call back processing is required due to the asynchronous behavior of distributed process interactions. A DLL call back is like an event notification to a user application signaling that a service request has been completed, or that the processing of received data is required. The helper file content is similar to the HLA Federation Execution Detail (FED) file, and since the APTT application is viewed by HLA as a simulation, it has been named the Simulation Execution Detail (SED) file. The SED file will comprise enumerated tables that define periodic polling requirements of the APTT application. It functions as the rulebook for IPC by defining the periodicity time values of APTT polling to the HLA_APT DLL.

An offline tool will be developed to configure the APTT application object model and to output the application specific SED file. The tool will be a Microsoft Foundation Class (MFC) type application that enables the developer to create the SED file used during application execution. The design approach for this tool models the HLA Object Model Development Tool (OMDT) and is called the HLA APTT Tool (HAT).

### 3.1.3 Marketing/Commercialization Approach

AAC's marketing approach is to establish a feasible design, and provide briefings to various government agencies and commercial entities. Collaboration with Cybernet is key to success. Cybernet has a solid history with SBIR efforts, and AAC will implement current Cybernet marketing strategies and products. Moreover, Cybernet is currently active in the market with similar products.

AAC plans to utilze the World Wide Web (www) for distrubution and sale of the SDK product. Licensing will be established to application developers as a function of the number of products sold that embed the SDK functionality. Annual maintenance and support agreements will be made available that provide subscribers updated releases of the SDK as well as technical support.

Upon Phase II award, AAC and Cybernet will establish a joint marketing and commercialization plan. Phase I collaboration with Cybernet has been limited, and its believed that when SBIR funding supports Cybernet's active participation, an existing Cybernet approach will be adopted to this effort.

# 4 Work Completed

## *4.1 Initial Design*

Table 3-1 below identifies work efforts performed during the first three months of performance. As indicated, work was suspeneded on the initial design effort due to the discovery of a COTS product that encapsolutated most of the software that was under development. Moreover, this COTS product provided an HLA complient network API that expanded the scope of the initial design to meet DoD requirements for distrubuted simulations – an APTT functional expansion expected in the future. The COTS product discovered through investigation is OpenSkies, developed by Cybernet Systems Corporation.

Table 3.1-1 Work Completion Dates

| Task | Planned Completion | Actual Completion |
|---|---|---|
| In Task Order: | | |
| 1 *Task I: Functional Requirements Analysis* | ------ | -------- |
| 2 T1-Build 1 | 12/24/99 | 12/23/99 |
| 3 T1-Build 2 | 1/27/00 | 1/25/00 |
| 4 T1-Build 3 | 4/26/00 | |
| 5 *Task II: Performance Requirements Analysis* | -------- | -------- |
| 6 T2-Build 1 | 1/3/00 | 1/6/99 |
| 7 T2-Build 2 | 2/7/00 | 2/8/00 |
| 8 T2-Build 3 | 5/10/00 | |
| 9 *Task III: System Design* | -------- | -------- |
| 10 T3-Build 1 | 1/11/00 | 1/13/00 |
| 11 T3-Build 2 | 2/17/00 | 2/18/00 |
| 12 T3-Build 3 | 5/18/00 | |
| 13 *Task IV: Prototype Design* | -------- | -------- |
| 14 T4-Build 1 | N/A | N/A |
| 15 T4-Build 2 | 2/24/00 | (suspended) |
| 16 T4-Build 3 | 5/31/00 | |
| 17 *Task V: Demonstration and Final Report* | -------- | -------- |
| 18 T5-Build 1 | 1/13/99 | 1/18/99 |
| 19 T5-Build 2 | 3/9/00 | (suspended) |
| 20 T5-Build 3 | 6/7/00 | |
| 21 Draft Final Report | 4/12/00 | |
| 22 Final Report | 6/14/00 | |
| In Scheduled Order: | | |
| 18 T5-Build 1 | 1/13/99 | 1/18/99 |
| 2 T1-Build 1 | 12/24/99 | 12/23/99 |
| 6 T2-Build 1 | 1/3/00 | 1/6/99 |
| 10 T3-Build 1 | 1/11/00 | 1/13/00 |
| 3 T1-Build 2 | 1/27/00 | 1/25/00 |
| 7 T2-Build 2 | 2/7/00 | 2/8/00 |
| 11 T3-Build 2 | 2/17/00 | 2/18/00 |
| 15 T4-Build 2 | 2/24/00 | (suspended) |
| 19 T5-Build 2 | 3/9/00 | (suspended) |
| 21 Draft Final Report | 4/12/00 | |
| 4 T1-Build 3 | 4/26/00 | |
| 8 T2-Build 3 | 5/10/00 | |
| 12 T3-Build 3 | 5/18/00 | |
| 16 T4-Build 3 | 5/31/00 | |
| 20 T5-Build 3 | 6/7/00 | |
| 22 Final Report | 6/14/00 | |

### *4.2 Current Design*

Mid way throught the third month of Phase I performance, AAC discovered OpenSkies. Due to the design change impacts of incorporating the OpenSkies system, work associated with designing software to implement existing OpenSkies functionality was suspended. The discovery of OpenSkies has markedly acelerated the planned effort as work can now procede to integration issues associated with APTT user applications and applicable OpenSkies system components.

## 5 Design

The Networked APTT requires an embedded infrastructure (layer of software) between all distributed APTT applications to support data sharing using the Internet. This layer of software is analogous to a phone system, whereby distributed applications can communicate and share information. The link, or API, between user applications and the networking software delivers user application required services to include Inter Process Control (IPC) and data protocol standardization.

This API must incorporate a defined protocol to ensure standardized communication between distributed user applications. Existing HLA software provides the required network services, and ensures standard protocols as defined in the HLA Object Model Template (OMT). The missing HLA capability is enabling APTT applications to function as HLA Simulation Objects. AAC's design approach facilitates the conversion of existing applications, and the development of new applications, to function as HLA Simulation Objects (Federates).

Products requiring design and development are identified to be:

(1) An APTT HLA API

(2) A tool to facilitate APTT application modification, development, and integration to this new HLA API.

### *5.1 APTT HLA API*

Existing APTT applications were not developed with HLA as a design objective. Moreover, APTT applications were (and will continue to be) developed by multiple vendors using a variety of software languages (Visual Basic, Authorware, Rapid, etc.). HLA is designed to work with C++ developed applications, and it closely mirrors the C++ Object Oriented Design methodology. Hence, non-C++ developed applications cannot utilize the current class based API and call back processes required for HLA integration. To support APTT application integration with the HLA environment, an

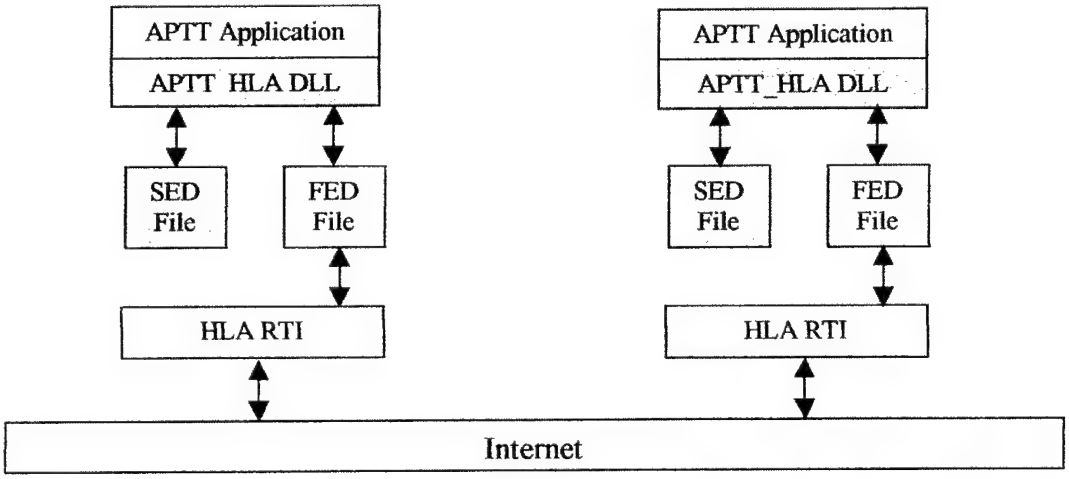
additional layer of software must act as the liaison between the APTT application and the current HLA API. In conjunction with the development of this new layer of software (API), an off line tool must be developed to enable vendors of APTT applications to extend current capabilities to interface with the new HLA API.

The AAC design requires the development of an APTT_HLA DLL to provide this required API. The APTT_HLA DLL encapsulates the two OpenSkies DLLs (Networking and Recording), and adds data buffering and distributed data IPC. This added APTT_HLA DLL functionality is required as many non-C++ developed APTT applications can only make "export C" function calls, and cannot process class based call backs. The OpenSkies Application is developed in C++ and this new functionality was not required for the Network and Recording DLLs. Hence, the APTT_HLA DLL will be extended to include an export C function interface to support all API requirements.

The HLA APTT Tool (HAT) will enable the APTT developer to define a user application initiated polling periodicity that APTT_HLA DLL export C functions will utilize to enable IPC processing.

The APTT_HLA DLL will require initialization from a data file comprising HAT generated output data. This file performs the same functionality for the APTT application as the FED file does for the HLA RTI in that it provides the APTT_HLA DLL with execution details associated with a particular APTT application. Since HLA would view the APTT application as a Simulation, this file will be called the Simulation Execution Detail – or SED file.

Following is a pictorial view of the APTT_HLA DLL component and its interfaces. Note that the APTT_HLA DLL is pulled into the execution space of a given APTT application making the APTT_HLA DLL an integral part of each APTT application. The APTT_HLA DLL and the SED File are shaded to identify them as the two components under design and development for this effort. Note that an APTT application can be any GUI based application to include training applications and instructor applications.



In summary, the APTT_HLA DLL is an innovative adaptation and extension of the OpenSkies Networking and Recording DLLs with added functionality to support an export C function interface as defined in the SED File.

### *5.2 HLA APTT Tool (HAT)*

In conjunction with HAT development, specifications and user documentation will be developed to define the process to conform or build APTT User Applications to function as HLA Simulations pursuant only to distributed data (public attributes and parameters) and data transfer contracts (interactions) between distributed applications.

The COTS Object Model Development Tool (OMDT), developed by AEgis Research, automates the process of creating HLA compliant Simulation Object Models (SOMs) and Federation Object Models (FOMs). For an APTT application to become a HLA object (federate), data structures and methods to be distributed are mapped into OMDT to create the application SOM. The SOMs (for each application), defines exported (published) and imported (subscribed) data (attributes) and distributed application interactions (IPC contracts). Multiple SOMs collectively create the FOM. Upon completion of the SOM and FOM in OMDT, OMDT creates (outputs) a Federation Execution Details (FED) file, which is used by the HLA RTI at runtime to initialize execution of a new federation. The FOM is the basis from which all federates in a given federation execution communicate. Moreover, the FOM umbrellas individual SOM attributes, parameters, and interactions for a given federation execution. Therefore, each SOM in the federation is fully represented in a given FOM.

To enable Federation compatible SOM development by APTT application vendors, a generic APTT Reference FOM (RFOM) must be defined that provides a standard data dictionary for all vendors of networked APTT appliations. Using HAT, the vendor can map existing (or planned) application data names, data types, and appication interactions to the RFOM. As new APTT applications are created, they will utilize applicable data from the RFOM. When new APTT applications require new data, the RFOM will be expanded to "umbrella" this new requirements. Hence, the RFOM expands to encompass API requirements for all APTT applications desiring to interact in an HLA federation.

HAT is a stand-alone tool separate from the OMDT. It can be viewed as a functional extension of OMDT, and will re-output current OMDT HLA products (HLA OMT tables) as well as APTT_HLA DLL required tables that support APTT HLA IPC. Moreover, HAT adds to the OMDT output the APTT_HLA DLL needed IPC information specific to each APTT application.

HAT incorporated OMDT functionality will include the configuration of all HLA Object Model Template (OMT) tables:

**Object Model Identification Table**
Metadata that describes the model and provides general information supporting reuse of the defined HLA Object.

**Object Class Structure Table**
Records the class relationships between varying types of federation objects. APTT applications built in a non-OOD environment will result in a planer (single column) table identifying functional elements of the application. Attributes being shared or needed by distributed applications can then be associated with this functional elements.

**Object Interaction Table**
Records the types of interactions that are possible between different application functional elements.

**Attribute Table**
Specifies details of federation scope (public) attributes for a given APTT application, and in the FOM.

**Parameter Table**
Specifies details of parameters passed during an interactions for a given APTT application and in the FOM.

**Enumerated Datatype Table**
Specifies datatypes that can be assigned only a finite set of values. Comprises a list of valid values. This table is a sub-component of the Attribute and Parameter tables.

**Complex Datatype Table**
Specifies datatypes that are aggregates of RTI-based types, enumerated types, and other complex types in a single structure. This table is a sub-component of the Attribute Table.

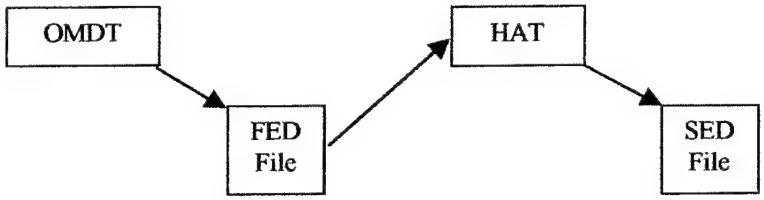
**Routing Space Table**
Provides a common mechanism for specifying the exchange of shared data and coordinating distributed data management among federate members. Attributes of table records define filters that control scheduling and activation of RTI services.

**FOM/SOM Lexicon**
Defines all the terms used in the tables.

Added HAT functionality will include configuration of all APTT applications to support export C function driven interactions. HAT outputs associated with interaction scheduling and data receipt notification will be compiled and output to a separate Simulation Execution Detail (SED) file. The federation scope of the SED is constrained

to the given APTT application for which it was tailored. Moreover, the SED exists to support application loaded APTT_HLA DLL execution. Processing associated with SED data is transparent to HLA Federation execution, as the HLA COTS software treats the APTT application as a standard HLA federate. The SED defines execution rules the APTT application executes to look and act like a standard HLA federate.

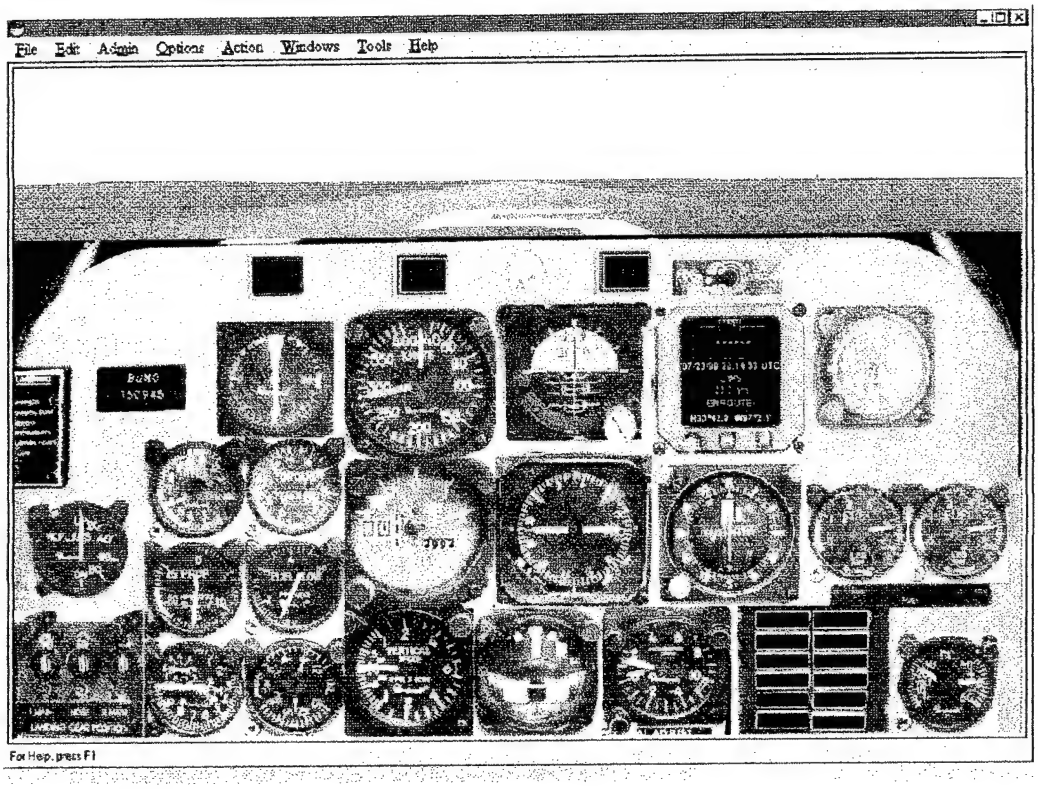
In Summary, the HAT is a repackaging of the OMDT output products with added capability to output a SED file. The HAT will be used sequentially after usage of the COTS OMDT. Moreover, the HAT will input the OMDT outputs to generate the needed SED file as shown below:




### 5.3 *Cybernet OpenSkies System*

The following supporting data from Cybernet Systems Corporation on OpenSkies describes current OpenSkies functionality. The software architecture of OpenSkies supports parting out the parent appliacation to provide the scoped functionality applicable to the Networked APTT.

Cybernet's OpenSkies Virtual Environment Training System will ease the development and broaden the capabilities for Avionics Part Task Training. Cybernet's System already has a HLA interface completely integrated into a virtual environment training system. This allows us to create scenarios for training pilot/copilot situations where the students reside at different stations in different locations and are able to operate control interfaces in a single aircraft. Further this system extends the capability to allow instructors to 'ride along' in the aircraft and monitor or take control of the aircraft. Cybernet's OpenSkies system has a complete Software Development Kit for creating new aircraft with realistic flight models and realistic panel displays. These panels can be created from actual bitmapped photographs of the instrument panels to allow for completely realistic operational controls.




The OpenSkies system is also a complete quantitative training performance measurement system. All of the data from the simulation is recorded for analysis and playback. This includes such things as changes in dials and switches, radio calls and manipulation of the aircraft. The performance measurement capability allows the system to grade the student and show the student where he/she has made mistakes or errors.

Overall this system provides an interface for creating interfaces and training scenarios for any type of Part Task Trainer as well as providing instructors with a CBT system for the quantitative measurement of student performance.

### 5.3.1 OpenSkies Training System

The topic objective is to design an inovative approach to networking APTT appliations. Incorporation of OpenSkies meets this objective, and provides the additional training system value inherant in the OpenSkies system to the APTT user.

#### 5.3.1.1 What does it do?

Many simulators have the capability to familiarize the student with simulations of the actual instruments and a few have the capability to create scripts for mission play. The OpenSkies Training system provides a training scenario authoring tool that allows for the training of both students and instructors in a Virtual Reality Environment for any type of simulation. The strength of the OpenSkies Training System relies on the close

integration of the analysis and performance measurement capabilities directly within the simulation software.

#### 5.3.1.2 Why do I need it?

OpenSkies is based on the training methodologies developed in the Naval Air Warfare Center Training Systems Division (NAWCTSD) for actual Navy training. OpenSkies was created with the following key ideas behind it.

- To improve the student's performance beyond current training program capabilities
- To provide a measurable performance standard
- To provide the fast, simple creation of new training courses
- To provide a low cost solution to training on complex, expensive equipment

#### 5.3.1.3 How will this system improve the students performance?

This system provides capabilities for:

- providing for a more quantitative approach to performance measurement
- showing students exactly where their deficiencies exist and allowing them to concentrate on those items
- providing an interface for creating personalized scenarios to address a student's particular weaknesses
- supporting student proficiency at specific levels of performance
- providing a more structured environment for teaching instructors

#### 5.3.1.4 What is unique about this system?

- Applies a quantitative approach that allows for a better comparison of performance.
- Provides for tracking of class level of performance
- Provides support of rating standardization of instructors
- Provides for the development of training scenarios in virtual reality simulations in a matter of hours rather than days or weeks
- This system may be customized to any domain for faster scenario development.

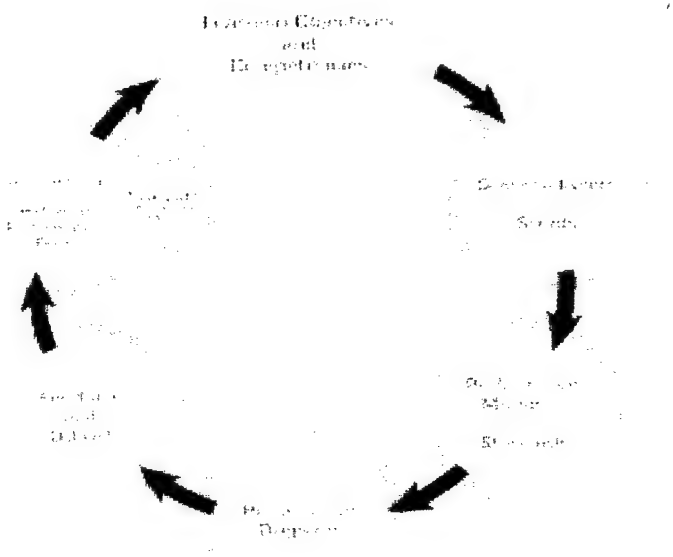
#### 5.3.1.5 What are the payoffs/advantages of this system?

- Applies structured event based training methodologies to a virtual environment
- Allows for recording and playback of the entire training mission for later analysis
- Provides automatic performance analysis feedback to the student
- Low hardware costs – supported by sub $1000 PCs
- Networked multi-participant capabilities for team training

### 5.3.2 OpenSkies Scenario Development and Performance Measurement

Many systems have been created for the development of realistic simulations. However, systems for creating effective learning environments are only beginning to emerge. One method that has been developed at the NAWCTSD is the Event-Based Approach to Training (EBAT). This method shows considerable promise and has been successfully used in a number of settings to establish effective learning environments. EBAT provides a systematic approach for developing learning objectives, generating scenarios, measuring performance, and providing feedback (Oser, Cannon-Bowers, Dwyer, and Salas, 1998).

Cybernet has worked with NAWCTSD through the Small Business Innovative Research program to create the OpenSkies Training System. This system has closely integrated the EBAT with an OpenGL based virtual environment toolkit to produce a training scenario authoring system. This authoring system employs the systematic approach to training developed by Fowlkes, Lane, Salas, Franz, and Oser (1994). This approach is diagramed below as an iterative procedure.



By using this approach, OpenSkies is able to provide an effective learning environment for students and instructors.

## 6 Positive / Negative Results

Initially it was presumed that an SDK had to be developed from scratch to achieve the Networked Avionics Part Task Trainer (APTT) topic objective. This presumtion was based on preliminary COTS research that failed to identify the needed product. Hence, requirements were established and proposed for this product for development of an SDK based on Microsoft's operating system level Winsock DLL.

However, the search continued for a COTS product that provided the needed functionality. One was found early in the third month of performance. The product discovered is OpenSkies, developed by Cybernet Systems Corporation under SBIR funding.

AAC also discovered the DoD mandate to use the High Level Architecture (HLA) environment for all simulation development. HLA developed simulations (Federates) inherently have networking capability to all other HLA developed simulations (Federates) operating in an HLA Federation. An HLA Federation requires that all Federates interface to the platform resident HLA Run Time Infrastructure (RTI) that provides the networking services to the Federates. OpenSkies is an HLA complient system.

Although APTT applications are training applications vs. simulations, they generate similar networking requirements. Moreover, simulation hosting on APTT platforms is expected in the future - as PC processing speeds increase and memory costs decrease.

The OpenSkies discovery is gross reduction in effort required to realize the end topic objective. The development time and risks associated with the initial Phase I design approach have been grossly reduced as OpenSkies slides in to meet needed functionality. Using the OpenSkies based SDK, existing APTT applications can be upgraded to HLA Federates networking in an HLA Run Time Infrastructure (RTI), and new APTT applications can incorporate the functionality during development.

The negative aspect of the OpenSkies discovery is that analysis and design work that has occured prior to discovering OpenSkies was counter productive – an efficient and better mouse trap already existed.

During month 3 of the Phase I effort, AAC established a non-disclosure agreement with Cybernet Systems Corporation to colaborate on Phase II, and to provide AAC with needed technical data to develop a conceptual design for Phase I. Post month 3 work did not realize colaboration with Cybernet regarding technical data on OpenSkies; thereby precluding AAC's ability to develop a detailed conceptual design. In leu of missing data, AAC exposed high level aspects of OpenSkies enabling the identification

of two OpenSkies components required to achieve an embedded HLA API for APTT user applications.

# 7 Phase II Plan

## Introduction

This document describes AAC's plans to build upon work accomplished under the SBIR N99-185 Phase I effort to design, implement, commercialize and market a COTS product to support design and development of networking Avionics Part Task Trainer (APTT) applications. Our plan includes a detailed description of Phase II objectives, work plan, anticipated benefits, transition plan, Financial plan, our qualifications for undertaking this effort, and estimate of program costs.

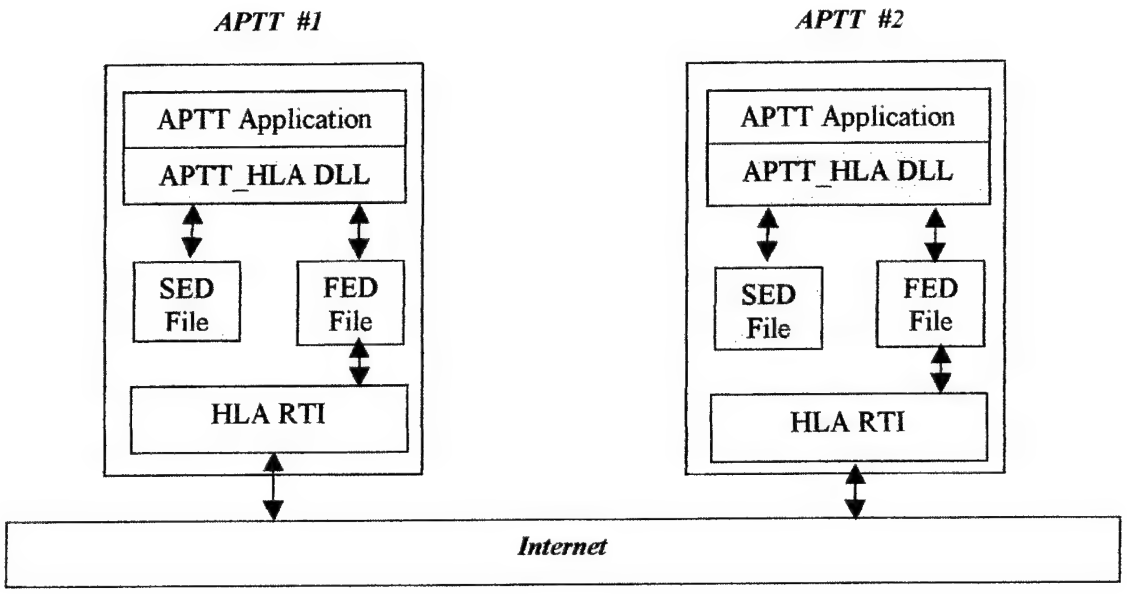
## Phase I Synopsis

AAC currently is nearing completion of the Phase I study on the Networked APTT. AAC initial proposed a build from scratch design concept centered on the Windows Winsock API. AAC also proposed to concurrently investigate and research industry for potential COTS products that provided similar applicable functionality.

During month three of Phase I performance, AAC discovered the OpenSkies product developed by Cybernet Systems Corporation. OpenSkies is an HLA compliant flight simulation application that provides collaborative distributed flight mission training. Two OpenSkies components were identified as meeting the networking requirements of the proposed design – the OpenSkies Network DLL and the Recording DLL.

AAC shifted from the proposed design to a new design incorporating these COTS components. Non-disclosure agreements were instantiated between AAC and Cybernet for Phase I data support and Phase II collaboration. AAC received high-level technical data on the two DLL components, as well as OpenSkies product information. From this data, AAC has defined the following design architecture for the Networked APTT. Shaded boxes depict the system components derived from the SDK product. The APTT_HLA DLL comprises an export C function interface to the user application. The SED file is a product of using the HLA APTT Tool, which is part of the SDK. The SED file contains Inter Process Control (IPC) management tables that enable the APTT_HLA DLL to act as mediator between the user application and the HLA RTI by way of the HLA FED file.

AAC is currently defining APTT_HLA DLL interface requirements and HAT generated SED file table formats. APTT application developers will need to go through this same process; hence, as part of the SDK, documentation will exist to assist in SED file definition (HAT user manual) and an ICD for each software application tool (DLL).




From this design architecture, AAC initiated and accomplished high-level design requirements for an SDK product to support vender development of APTT applications with networking functionality.

APTT applications are currently developed by multiple vendors using a variety of Integrated Design Environments (IDEs). A primary goal in this SDK development is to ensure that it contains applicable high productivity tools addressing multiple IDEs such as C/C++, Visual Basic, Authorware, Quest, Tool-Book II, and Rapid CBT.

**Phase II Objective**

The objective of Phase II is to Implement and validate a prototype Networked APTT system developed using the SDK product. The prototype system will utilize the AH-1W Control Display and Navigation Unit Emulation (CDNUE) as a sample of APTT applications. The prototype will demonstrate that the SDK product enables APTT application development and modification to perform collaborative functionality between distributed APTTs using the Internet.

AAC's objective is to demonstrate the feasibility of the product SDK through demonstration of this prototyped APTT application to function as a PC hosted HLA federate. Modified (prototyped) CDNUEs will be installed on two distributed APTTs, and an HLA federation will be launched. The distributed CDNUEs will then demonstrate collaborative functionality associated with ARC-210 radio communications.

This prototype system will provide Networked APTT proof of concept and support demonstrations in marketing activities.

**Work Plan**

AAC's formal Phase II work plan will be established jointly by AAC and Cybernet. Detailed knowledge of the two OpenSkies components is required to properly program modifications, component integration, and new software requirements and designs.

The work plan will comprise a spiral build development philosophy. The overlying design will be established, and a single thread will be implemented as a prototype making design adjustments as necessary. Upon prototype development, The AAC/Cybernet team will revisit, verify or modify initial design requirements, and add to the core software established with the prototype. Each planned build will initiate upon completion of the prior build and will encompass requirements definition, design, implementation, and test processes.

**Anticipated Benefits:**

AAC's teaming with Cybernet brings product development and commercialization experience to this SBIR contract. The government has awarded Cybernet Systems Corporation multiple Phase I and Phase II contracts on SBIR topics N94-027, AF96-022, and OSD97-004. These contracts resulted in the realization of the current state of OpenSkies. Leveraging the Networked APTT effort against this established software system reduces development risk and increases the government's return on investment.

Demonstration of a working prototype will provide substantial proof that the merits of collaborative training required with today's avionics systems is a cost effective approach to complex multi-aircrew avionics (Navigation and Communication) related processes.

The use of avionics simulations on APTTs is expected to increase as PC performance improves and costs drop. APTT avionics simulations represent a subsystem simulation with respect to an overall aircraft avionics suite. HLA compliant subsystem simulations are not only useful in the APTT, but can be migrated to advanced environments like full aircraft simulations and weapon system trainers based on HLA reuse. This topic solution represents a marked potential for increased return on investment - HLA compliant APTT type simulations can be re-used by larger trainer systems. Moreover, as aircraft avionics systems receive Operational Flight Program (OFP) upgrades and enhancements, the APTT will become the target development platform for avionics representative subsystem simulations. Once developed, tested and verified, these APTT simulations can be "rolled up" to the larger training systems as plug-and-play HLA compliant components.

The need for distributed processing with APTTs will also become more prevalent as coordinated aircrew missions rely increasingly on their avionics system to facilitate execution. Networked APTTs enabled by AAC SDK will support planning and executing

federations focused on specific coordinated mission scenarios. Communication is the key to Command and Control, and the Networked APTT provides cost effective training using the complex radio systems emerging in aircraft today.

**Transition Plan**

AAC plans to team with Cybernet Systems Corporation for Phase II execution to prototype the Networked Avionics Part Task Trainer environment based on a derivative of their OpenSkies Software.

The current OpenSkies software system is the result of multiple SBIR funded programs. The OpenSkies product was leveraged from the T-TACTS N94-027, the Force Feedback Virtual Fixture and Reality Registration for Mid-Air Refueling AF96-022, and Commercial Game Development Using HLA and SDRIS SBIR topics. Cybernet is currently investing heavily for OpenSkies to enter the commercial market. The Networked APTT SDK is planned to enter all potential markets along with OpenSkies.

Marketing and distribution of this product will be coincident with marketing and distribution activities currently associated with OpenSkies commercialization. The Networking APTT SDK may be made available to the market as a subset of the OpenSkies SDK, and distributed by way of Internet downloading and licensing.

AAC plans to team with Cybernet during the development and commercialization of this APTT SDK. As a Phase II team member, Cybernet will provide their current knowledge base and product expertise associated with HLA and their existing OpenSkies product. AAC will design and develop the APTT networking SDK using AAC developed tools and processes along with the tools and OpenSkies software components provided by Cybernet to develop the prototype Networked Avionics Part Task Trainer. Joint AAC and Cybernet development work will include innovative development and modifications of Cybernet's tools and software to meet product development requirements.

AAC and Cybernet intellectual property issues associated with the Networked APTT will be addressed after Phase II award. However, it is known that OpenSkies un-related software will be retained as Cybernet Intellectual Property. AAC intellectual property will encompass 100% of the HAT tool, and a percentage of the APTT_HLA DLL. Details are yet to be worked out regarding the percentages.

**Financial Plan**

AAC is a small private company who will require additional funding and expertise to achieve a prototype within the two year Phase II performance period. AAC is eager to team with Cybernet who have committed to work with AAC. Cybernet brings to the table applicable expertise and tools required to realize this topic Phase II objective.

AAC's corporate mission is to develop networked PC based trainers such as the APTT. AAC will continue to invest IR&D dollars to advance the objective of prototyping the Networked APTT independently of SBIR funding.

Internal funding over the ensuing months are considered to be matching dollars to funding provided by a Phase II award. Internal AAC investment in IR&D has been limited, and will "ramp up" upon completion of Phase I during the foreseen gap between Phase I and Phase II award.

AAC has marketed US Airways and has received positive feedback that they are interested in the commercialization of this networking SDK for APTTs. In March 1999, US Airways stated that they were directed that no new outsourcing could occur. This dampened AAC's marketing efforts with them considerably. With the now anticipated acquisition of US Airways by United Airways, AAC optimistically expects to resume marketing with these same key individuals.

### AAC Qualifications

Although this SBIR Phase I effort is AAC's initiation into the SBIR program, AAC has an extensive history with avionics systems. AAC's president, and prime investigator for this topic, has earned a Baccalaureate degree in Computer Science and is currently pursuing a Masters degree in Computer Science with a focus on network systems development.

Cybernet has proven capability to develop distributed training systems as is evident in the OpenSkies distributed flight training system. Moreover, Cybernet has a successful history with multiple SBIR contracts. Current Cybernet IR&D efforts include network optimization of Protocol Delivered Units (PDUs) to minimize data through put for real time distributed simulations; thereby minimizing network hardware bandwidth requirements.

AAC and Cybernet are independently pursuing similar networked trainer objectives making a powerful team on this effort for Phase II.

## 8 Conclusions and Recommendations

APTT applications have been (and will be) developed by multiple vendors in proprietary development environments. Hence, an SDK that enables conversion of existing APTT applications and the development of new Networked APTT applicatoions must provide tools that are compatible with multiple Integrated Development Environments (IDEs) to include C/C++, Visual Basic, Authorware, Quest, Tool Book II, and Rapid CBT.

AAC concludes that OpenSkies provides cost effective leveraging to the solution of meeting the networking requirements associated with this topic objective. AAC

recommends using OpenSkies combined with innovative software for the creation of Networked APTT applications.

AAC recommends that DoD establish a bottom's up upgrade and development plan associated with all avionics simulation software development using the APTT as the development platform. Reuse of HLA compliant APTT applications in differing computer environments is inherent in HLA. Therefore once an APTT simulation (perhaps imbedded in an APTT CBT application) is developed to function as an HLA Federate on a PC hosted RTI, it can be migrated to a Weapon System Trainer that may use a Sun Workstation hosted HLA RTI.